

Task - 6

Program 1:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
public class StudentDatabase {
private      static      final      String      JDBC_URL      =
"jdbc:mysql://localhost:3306/student_database";
private static final String USERNAME = "your_username";
private static final String PASSWORD = "your_password";
public static void main(String[] args) {
String createTableSQL = "CREATE TABLE students ("
+ "id INT AUTO_INCREMENT PRIMARY KEY,"
+ "name VARCHAR(100),"
+ "age INT,"
+ "grade VARCHAR(10)"
+ ")";
String insertDataSQL = "INSERT INTO students (name, age, grade) VALUES "
+ "('John Doe', 20, 'A'),"
+ "('Jane Smith', 21, 'B'),"
+ "('Mike Johnson', 22, 'C)";
try (Connection connection = DriverManager.getConnection(JDBC_URL,
USERNAME, PASSWORD);
Statement statement = connection.createStatement()) {
statement.execute(createTableSQL);
System.out.println("Table created successfully.");
statement.executeUpdate(insertDataSQL);
System.out.println("Data inserted successfully.");
} catch (SQLException e) {
e.printStackTrace();
}
}
}
```

Program 2:

```
import java.sql.*;
public class StudentDatabase {
    static final String JDBC_URL = "jdbc:mysql://localhost:3306/student_db";
    static final String USERNAME = "username";
    static final String PASSWORD = "password";
    public static void main(String[] args) {
        try (Connection connection = DriverManager.getConnection(JDBC_URL,
            USERNAME,
            PASSWORD)) {
            createTable(connection);
            insertData(connection, "John Doe", 25, "Computer Science");
            updateData(connection, "John Doe", 26);
            deleteData(connection, "John Doe");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    private static void createTable(Connection connection) throws SQLException {
        String sql = "CREATE TABLE IF NOT EXISTS students (" +
            "id INT AUTO_INCREMENT PRIMARY KEY," +
            "name VARCHAR(255) NOT NULL," +
            "age INT NOT NULL," +
            "major VARCHAR(255) NOT NULL)";
        try (Statement statement = connection.createStatement()) {
            statement.execute(sql);
        }
    }
    private static void insertData(Connection connection, String name, int age,
        String major) throws
        SQLException {
        String sql = "INSERT INTO students (name, age, major) VALUES (?, ?, ?)";
        try (PreparedStatement preparedStatement =
            connection.prepareStatement(sql)) {
            preparedStatement.setString(1, name);
        }
    }
}
```

```

preparedStatement.setInt(2, age);
preparedStatement.setString(3, major);
preparedStatement.executeUpdate();
}
}
private static void updateData(Connection connection, String name, int
newAge) throws
SQLException {
    String sql = "UPDATE students SET age = ? WHERE name = ?";
    try          (PreparedStatement          preparedStatement          =
connection.prepareStatement(sql)) {
        preparedStatement.setInt(1, newAge);
        preparedStatement.setString(2, name);
        preparedStatement.executeUpdate();
    }
}
private static void deleteData(Connection connection, String name) throws
SQLException {
    String sql = "DELETE FROM students WHERE name = ?";
    try          (PreparedStatement          preparedStatement          =
connection.prepareStatement(sql)) {
        preparedStatement.setString(1, name);
        preparedStatement.executeUpdate();
    }
}
}

```